| Project | **IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>** |
|---|---|
| Title | **Text and tables to resolve ballot #4a comments** |
| Date Submitted | **2002-02-21** |
| Source(s) | Subbu Ponnuswamy       Voice: 916-425-1276<br>1005 Blue Ravine Road, # 926    Fax:   413-521-2612<br>Folsom, CA 95630           mailto:subbu@acm.org |
| Re: | Re-circulation ballot #4a |
| Abstract | Text and tables accompanying ballot #4a. |
| Purpose | |
| Notice | This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate text contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16. |
| Patent Policy and Procedures | The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard."<br><br>Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>. |

# 1   Specific Changes to P802.16a/D2-2002 - #1

Add the following text to page 23, line 29:

## 6.2.2.2 MAC sub-headers

*Replace clause 6.2.2.2.1 with the following*

### 6.2.2.2.1 Fragmentation sub-headers

The Fragmentation sub-header (FSH) for non-ARQ connections is shown in Table 7. The Fragmentation sub-header for ARQ connections is a simple extension of the non-ARQ Fragmentation sub-header. The fragmentation sub-header for ARQ connections is shown in Table 8.

*Replace Table 7 with the following*

**Table 7:  Non-ARQ  Fragmentation  Sub-header  Format**

| Syntax | Size | Notes |
|---|---|---|
| Non-ARQ Fragmentation sub-header () { | | |
| FC | 2 bits | |
| FSN | 3 bits | |
| *Reserved* | 3 bits | |
| } | | |

*Insert Table 8 and the following text*

**Table 8:  ARQ  Fragmentation  Sub-header  Format**

| Syntax | Size | Notes |
|---|---|---|
| ARQ Fragmentation sub-header () { | | |
| FC | 2 bits | |
| FSN | 3 bits | Bits 2:0 |
| *Reserved* | 3 bits | |
| FSN | 8 bits | Bits 10:3 |
| } | | |

The fields of the Non-ARQ Fragmentation sub-header are defined in Table 9. The fields of the ARQ fragmentation sub-header are defined in Table 10.

*Replace the caption of Table 8 with the following*

**Table 9:  Non-ARQ  Fragmentation  Sub-header  Fields**

*Insert Table 10*

**Table  10:  ARQ  Fragmentation  Sub-header  Fields**

| Name | Length | Description |
|---|---|---|

| FC | 2 bits | Fragmentation Control<br>Indicates the fragmentation state of the payload:<br>00 = no fragmentation<br>01 = last fragment<br>10 = first fragment<br>11 = continuing (middle) fragment |
|---|---|---|
| FSN | 11 bits | Fragmentation Sequence Number<br>Defines the sequence number of the current SDU fragment. This field increments by one<br>(modulo 2048) for each fragment, including un-fragmented SDUs. |

*Replace clause 6.2.2.2.3 with the following*

**6.2.2.2.3 Packing sub-headers**

When Packing (see 6.2.3.4) is used, the MAC may pack multiple SDUs into a single MAC PDU. When packing variable-length MAC SDUs, the MAC precedes each one with a Packing sub-header. The Packing sub-header for non-ARQ connections is defined in Table 13. The Packing sub-header for ARQ connections is a simple extension of the non-ARQ Packing sub-header. The Packing sub-header for ARQ connections is defined in Table 14.

*Replace Table 11 with the following*

**Table 13: Non-ARQ Packing Sub-header Format**

| Syntax | Size | Notes |
|---|---|---|
| Non-ARQ Packing sub-header () { | | |
| FC | 2 bits | |
| FSN | 3 bits | |
| Length | 11 bits | |
| } | | |

*Insert Table 14*

**Table 14: ARQ Packing Sub-header Format**

| Syntax | Size | Notes |
|---|---|---|
| ARQ Packing sub-header () { | | |
| FC | 2 bits | |
| FSN | 3 bits | Bits 2:0 |
| Length | 11 bits | |
| FSN | 8 bits | Bits 10:3 |
| } | | |

*Insert the following text*

The fields of the non-ARQ packing sub-header are defined in Table 15. The fields of the ARQ packing sub-header are defined in Table 16.

*Replace the caption of Table 12 with the following*

**Table 15: Non-ARQ Packing Sub-header Fields**

**Table 16: ARQ Packing Sub-header Fields**

| Name | Length | Description |
|---|---|---|
| FC | 2 bits | Fragmentation Control<br>Indicates the fragmentation state of the payload:<br>00 = no fragmentation<br>01 = last fragment<br>10 = first fragment<br>11 = continuing (middle) fragment |
| FSN | 11 bits | Fragmentation Sequence Number<br>Defines the sequence number of the current SDU fragment. This field increments by one<br>(modulo 2048) for each fragment, including un-fragmented SDUs. |
| Length | 11 bits | The length in bytes of the MAC SDU or SDU fragment, including the three-byte packing sub-header. |

*Insert clause 6.2.2.2.4*

**6.2.2.2.4 ARQ Feedback**

Note that the ARQ Feedback is not a separate sub-header. Whenever the TYPE value indicates the presence of ARQ Feedback the feedback information is transported as a payload, as described in ARQ and packing clauses. The TYPE values are used to indicate the presence or absence of ARQ Feedback information and do not result in the addition of any sub-headers.

# 2  Specific Changes to P802.16a/D2-2002 - #2

Remove clause 6.2.3.4.3 and add the following two clauses (6.2.3.4.2 and 6.2.3.4.3):

**6.2.3.4.2 Packing for ARQ connections**

The use of packing sub-header for ARQ connections is similar to that of non-ARQ connections as described in clause 6.2.3.4.1, except that the ARQ connections use a three-byte packing sub-header with an extended FSN. In addition, the fixed-length packing option (6.2.3.4.1.1) is not supported by ARQ connections. If packing is turned on for a connection, the MAC may pack multiple MAC SDUs into a single MAC PDU. The transmitting side has full discretion whether or not to pack a group of MAC SDUs and/or fragments in a single MAC PDU. Depending on the ARQ policies, the transmitter may choose to pack multiple fragments of the same SDU into a single MAC PDU, even if there is sufficient bandwidth to send the whole MAC PDU un-fragmented. While this does not change the semantics of the packing, the ARQ protocol may utilize this feature to allow flexibility in retransmission.

**6.2.3.4.2.1 Packing variable-length MAC SDUs**

The packing of variable-length MAC SDUs for the ARQ connections is similar to that of non-ARQ connections, when fragmentation is not enabled. The ARQ connections shall use the three-byte ARQ packing sub-header defined in Table 16. The FSN of the packing sub-header shall be used by the ARQ protocol to identify and retransmit lost PDUs. The primary difference between ARQ and non-ARQ packing is in the interaction with fragmentation, as described in the following sub-clause.

**6.2.3.4.2.1.1 Interaction with fragmentation**

Similar to the non-ARQ case, when a Packing sub-header is present, the fragmentation information for individual MAC SDUs or MAC SDU fragments is contained in the corresponding Packing sub-header. If no Packing sub-header is present, the fragmentation information for individual MAC SDU fragments is contained in the corresponding Fragmentation sub-header. Figure 136 shows the generic structure of a MAC PDU with various sub-headers.

| Generic MAC Header | Other sub-headers | ARQ/Non-ARQ Packing or Fragmentation sub-header | Payload (packed or otherwise) | CRC-32 |
|---|---|---|---|---|

**Figure 136 — Example MAC PDU with sub-headers**

Figure 137 illustrates the use of ARQ Fragmentation sub-header without packing.

| Generic MAC Header | ARQ Fragmentation Sub-header | Payload (One SDU or fragment of an SDU or ARQ Feedback IEs) | CRC-32 |
|---|---|---|---|

**Figure 137 - MAC PDU with ARQ Fragmentation sub-header**

Figure 138 shows the structure of a MAC PDU with ARQ packing sub-headers. Each of the packed MAC SDU or MAC SDU fragments or ARQ feedback payload requires its own packing sub-header and some of them may be transmissions while others are re-transmissions.
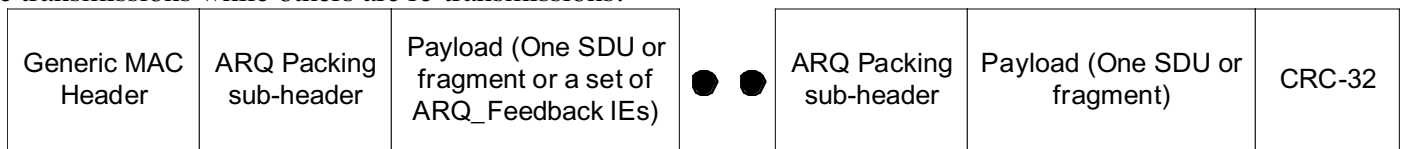
| Generic MAC Header | ARQ Packing sub-header | Payload (One SDU or fragment or a set of ARQ_Feedback IEs) | ● ● | ARQ Packing sub-header | Payload (One SDU or fragment) | CRC-32 |
|---|---|---|---|---|---|---|

**Figure 238 — Structure of MAC PDU with ARQ packing sub-header**

Unlike the non-ARQ case, it is possible to have continuation fragments packed with other fragments in the same MAC PDU for various reasons. For example, the ARQ mechanism may choose to fragment a MAC PDU into multiple fragments and pack them into the same MAC PDU during the first transmission, in order to support flexible re-transmission. Similarly, a MAC PDU may have fragments from different SDUs, including a mix of first transmissions and retransmissions. The 11-bit FSN and 2-bit FC fields uniquely identify each fragment or un-fragmented SDU.

# 3   Specific Changes to P802.16a/D2-2002 - #3

### 6.2.3.4.2 Packing ARQ Feedback payload
An ARQ feedback payload consists of one or more ARQ Feedback Information Elements (clause 6.2.4.2). ARQ feedback payload may be sent on an ARQ or non-ARQ connection. However, policies based on implementation and/or QoS constraints may restrict the use of certain connections for transporting ARQ feedback payload. The ARQ feedback payload is treated like any other payload (SDU or fragments) from the packing perspective, except that only one ARQ feedback payload shall be present within a single MAC PDU.

The presence of ACK Feedback payload in a MAC PDU is indicated by the value of the TYPE field in the generic MAC header (No sub-headers are added). When the message type indicates the presence of ARQ Feedback payload (e.g., types 0x0A and 0x0C), the first packed payload shall be the ARQ Feedback payload. The packing sub-header preceding the ARQ feedback payload indicates the total length of the payload including the two or three-byte packing sub-header and all ARQ Feedback Information Elements within the payload. The FSN field of the packing sub-header shall be ignored for the ARQ Feedback payload and the FC bits shall be set to 00.

# 4   Specific Changes to P802.16a/D2-2002 - #4
Replace Table 171 with the following

**Table 171: ARQ Feedback Information Element**

| Syntax | Size | Notes |
|---|---|---|
| ARQ_feedback_IE () { | Variable | |
| **CID** | 16 bits | The ID of the connection being referenced |
| **LAST** | 1 bit | 0 = More ARQ feedback IE in the list<br>1 = Last ARQ feedback IE in the list |
| **ACK Type** | 2 bits | 00 = Selective ACK entry<br>01 = Cumulative ACK entry<br>10 = Cumulative with Selective ACK entry<br>11 = Reserved |
| FSN | 11 bits | Fragment sequence number for the acknowledged ARQ fragment |
| **Number of 16 bits ACK Maps** | 2 bits | 00 = 16 bits<br>01 = 32 bits<br>10 = 48 bits<br>11 = 64 bits<br><br>Note: If ACK Type = 00 or 10, the bits have the meaning shown, but if ACK Type = 01, the field is reserved and set to 00 |
| If  (ACK Type != 01)<br>for (i=0; i< (Number of 16 bits ACK Maps   +1); ++i) { | | |
| **ACK MAP** | 16 bits | Each bit set to one indicates the corresponding ARQ fragment has been received without errors. The bit corresponding to the FSN value in the IE, is the most significant bit of the first 16-bit map entry. The bits for succeeding fragment numbers are assigned left-to-right (msb to lsb) within the map entry.<br><br>If the ACK Type is 10, then the most significant bit of the first 16-bit map entry must be set to one and the IE shall be interpreted as a cumulative ACK for the FSN value in the IE. The rest of the bitmap shall be interpreted similar to ACK Type 00. |
| } | | |
| } | | |